

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-21963

(P2004-21963A)

(43) 公開日 平成16年1月22日(2004.1.22)

(51) Int. Cl.<sup>7</sup>

G06F 17/60

G06F 17/30

F 1

G06F 17/60 314

G06F 17/60 124

G06F 17/30 170Z

テーマコード (参考)

5B075

審査請求 未請求 請求項の数 2 書面 (全 6 頁)

(21) 出願番号 特願2002-210284 (P2002-210284)  
(22) 出願日 平成14年6月15日 (2002. 6. 15)(71) 出願人 502260856  
鈴木 康  
神奈川県相模原市東大沼3-21-15  
(72) 発明者 鈴木 康  
大阪府枚方市中宮北町10-12-102  
Fターム(参考) 5B075 KK07 ND20 NK10 PP30 PQ02  
QM05 QS20 UU40

(54) 【発明の名称】 検索している会員からの距離で他会員を検索できる出会い系サイトの検索システム

(57) 【要約】 (修正有)

【課題】 出会い系サイトで、会員に他会員の住所そのものは伝えず、検索する会員との距離が希望の範囲の会員を検索結果として示す検索システムを提供する。

【解決手段】 郵便番号と緯度経度を項目に持つデータベース（緯度経度テーブル）をサーバにおく。各会員に、入会時に少なくとも郵便番号を入力させ、前記緯度経度テーブルより求めた緯度経度を会員のデータベースに加える。検索時に会員の緯度経度より会員間の距離を求め、希望する距離の範囲dに入る会員を紹介する。

【選択図】 図4

```

sub d(
local($f1,$f2,$g1,$g2,$f1r,$f2r,$g1r,$g2r,$n1,$n2,$a,$e,$r,$w);
$a=6378136.0;
$e=0.006694470;
$pi=3.1415926535897;
$f1r= $f1*$pi/180.0;
$f2r= $f2*$pi/180.0;
$g1r= $g1*$pi/180.0;
$g2r= $g2*$pi/180.0;
$n1=$a/(sqrt(1.0-$e*sin($f1r)*sin($f1r)));
$x1=$n1*cos($f1r)*cos($g1r);
$y1=$n1*cos($f1r)*sin($g1r);
$z1=$n1*(1.0-$e)*sin($f1r);
$n2=$a/(sqrt(1.0-$e*sin($f2r)*sin($f2r)));
$x2=$n2*cos($f2r)*cos($g2r);
$y2=$n2*cos($f2r)*sin($g2r);
$z2=$n2*(1.0-$e)*sin($f2r);
$r=sqrt(($x1-$x2)*($x1-$x2)+($y1-$y2)*($y1-$y2)+($z1-$z2)*($z1-$z2));
$w=sin(atan2($z2/$r,atan2($x2/$r,$y2/$r)));
return (int($a*2.0*$w/1000));
)

```

**【特許請求の範囲】****【請求項 1】**

事前にデータベースに全会員の郵便番号を入力しておき、検索する会員に他会員の住所を提示することなく、両者の距離を提示する出会い系サイトの検索システム。

**【請求項 2】**

郵便番号と緯度経度の対応表、および 2 地点の緯度経度から距離を計算する公式を用いる請求項 1 に記載の出会い系サイトの検索システム。

**【発明の詳細な説明】****【0001】****【発明の属する技術分野】**

10

この発明はインターネットにおける出会い系サイトの情報検索に関し、特に検索する会員に、他会員の身元情報を提示しないで、条件を満足する他会員を検索する方法に関するものである。

**【0002】****【従来の技術】**

近年インターネットの発達によりさまざまな情報検索システムが開発されている。本発明は、主として出会い系サイト（結婚情報サービスの情報検索などにも適用される）の情報検索に関するものである。出会い系サイトでは、各人がインターネットを通して、通常独身の男女が自分の個人情報（住所、電話、メールアドレス、学歴、収入など）を登録して会員になる。そして、住所、電話、メールアドレスなどのその個人を特定できるいわゆる身元情報を他会員（異性）には伏せておき、学歴、収入、都道府県名などの条件から各会員が他会員を検索し、メール交換などから交際などを始める趣旨の会である。

20

**【0003】****【発明が解決しようとする課題】**

会員同士が教えあうまでは、検索システムが各会員の身元情報を他会員に教えることは無い。例えば、住所に関しては、検索者には、他会員の住所ではなく、地方名（関東、九州など）や都道府県名など、住所を曖昧にした情報のみが検索結果として得られるようになっている。しかしながら、都道府県の境界や広い都道府県にいる会員にとっては、精度のよい情報が得られないばかりか、実際にすぐそばに住んでいるのに、住所の都道府県が異なるという理由だけで、会員との交流が妨げられることすらある。検索する会員が真に欲しい情報は、それぞれの会員が自分から何キロ離れて住んでいるかである。実際、出会い系サイトの会員にとって、出会う便利さや結婚してから実家に近いかが重要である。したがって、両者の距離を条件にして検索を行なうシステムが利用できればありがたい。

30

**【0004】**

現在の出会い系サイトで、距離が何キロ以内かで会員を探索するシステムは皆無である。従来は、会員の身元情報を曖昧にして検索する会員に提示するという発想の域を出ていなかったのかもしれない。各会員の住所だけでなく、検索する会員の住所も同時に用いて距離を計算するという概念が従来無かったというのが適当かもしれない。また、検索する会員に距離を提示しても、その検索される会員の身元情報を提示したことにはならないことに気がつかなかったのかもしれない。

40

**【0005】**

あるいは、両者の距離を提示するというアイデアがどこかに存在していたとしても、収益に見合う実現方法が考案できなかったということが考えられる。国土地理院や地理情報提供者などからデータベースを購入し、大規模な計算機を運用すれば実現は不可能ではない。ただ、そのようなデータベースは高価であり、また大規模なデータベースを運用するのに莫大な計算機管理コストがかかる。

**【0006】**

出会い系サイトは、大半が在宅ビジネスとして営業されている。したがって、地理情報提供者から高価なデータベースを購入することは不可能に近い。そこで、低価格でこの処

50

理を実現できれば、距離情報を提示する出会い系サイトの開発が可能となる。また、ホームページを運用する場合、計算機のサーバをレンタルする（ディスクサイズで料金が大きく異なる）ことになるので、データベースのサイズが大きいと、レンタルコストが大きくなり、その業者は実現を断念することにもなりかねない。したがって、本発明の課題は、データベースの入手コストが低いことと、データベースが小さな記憶容量に収まるかどうかである。

【0007】

【課題を解決するための手段】

上記課題を解決するために、本発明は以下のような手順をとる。

1. 郵便番号と緯度経度を項目にもつデータベース（緯度経度テーブル）を購入し（1 10  
万円前後、業務用ではなく、個人用データベース）、サーバにおく。
2. 各会員は、入会時に少なくとも郵便番号を入力する。その際に、緯度経度テーブルを用いて、緯度経度を計算し、それを会員のデータベース（会員テーブル）の項目に加える。会員テーブルは、各会員の入会時、住所変更時のみしか用いない。
3. 各会員が検索をする際には、検索条件として自分の住所から直線距離が何キロ以内かという範囲を与える。情報検索システムは会員テーブルの緯度経度（ $x_1, y_1$ ）,（ $x_2, y_2$ ）, . . .（ $x_n, y_n$ ）とその検索する会員の緯度経度（ $x_0, y_0$ ）から、（ $x_0, y_0$ ）と（ $x_1, y_1$ ）の対、（ $x_0, y_0$ ）と（ $x_2, y_2$ ）の対、. . . ,（ $x_0, y_0$ ）と（ $x_n, y_n$ ）の対のそれぞれについて距離を求める。
4. 緯度経度の対（ $x_a, y_a$ ）と（ $x_b, y_b$ ）から両者の距離は一意に決まる。計 20  
算式については後述する。
5. 情報検索システムは、検索者が事前に与えた距離の範囲にある会員を検索結果とする（他に条件があればその条件との論理積、論理和となる）。また、その出力された会員と検索者との間の距離を提示する。ただし、他会員の住所は決して提示しない。

【0008】

経度緯度の対から距離を計算する手順を以下に説明する。2地点の緯度経度（ $\phi_1, \lambda_1$ ）,（ $\phi_2, \lambda_2$ ）から両者の地表面距離  $S$  を計算する手順は以下で与えられる。直交座標地を（ $x_1, y_1, z_1$ ）,（ $x_2, y_2, z_2$ ）とすると、直距離  

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$
 が計算できる。ここで、各座標値は、 $N_1 = R / \sqrt{1 - e^2 \sin^2 \phi_1}$ 、 $N_2 = R / \sqrt{1 - e^2 \sin^2 \phi_2}$ 、ただし、 $R = 6,378,136 \text{ m}$ （地球の赤道半径）、 $e = 0.0167$ （地球の離心率）として、

$$x_1 = N_1 \cdot \cos \phi_1 \cos \lambda_1, \quad x_2 = N_2 \cdot \cos \phi_2 \cos \lambda_2$$

$$y_1 = N_1 \cdot \cos \phi_1 \sin \lambda_1, \quad y_2 = N_2 \cdot \cos \phi_2 \sin \lambda_2$$

$$z_1 = N_1 \cdot (1 - e^2) \sin \phi_1, \quad z_2 = N_2 \cdot (1 - e^2) \sin \phi_2$$
 また、半射程角（地球の中心から2地点を見込んだ中心角の2分の1の角） $\Psi$  は次の式から求められる。

$$\Psi = \sin^{-1} \left( (Rn/2) / N \right), \quad N = (N_1 + N_2) / 2$$

そして、半射程角（単位はラジアン） $\Psi$  が求まると、2地点間の地表面距離  $S$  は次式で与えられる。

$$S = 2 \cdot N \cdot \Psi$$

若干煩雑ではあるが、計算機による計算時間は会員数が数万程度であれば、實際上無視できる。地球の丸みを考慮している（地表面距離）が、両地点での標高は0であると仮定している。標高が高いところでは誤差が大きくなるが、地球の丸みを考慮していない（直距離）場合より精度がよい。

【0009】

【発明の実施の形態】

以下、データベースの2個のテーブル（図1、図2）を用いて本発明の実施の形態について説明する。まず、購入した郵便番号と緯度経度を含むデータベースに他の項目があったとしても、郵便番号、緯度、経度の3項目のみを抽出してテーブルを作成し（図1）、サ

10

20

30

40

50

一バに格納する。郵便番号は7桁であり、存在しない郵便番号を除いても、5, 000, 000レコードある。各レコードは、郵便番号の他に緯度8桁、経度8桁で、全レコードを格納すると、それだけで100メガバイト以上にもなる。もし、出会い系サイトだけの用途であれば、郵便番号の最初の5桁だけの情報で近似してサーバに記憶すれば、その100分の1のディスク領域(1メガバイト程度)ですむことになり、レンタルサーバ代は大きくはならない。最初の5桁だけで近似しても、誤差はおおむね5キロ以下におさまる。このことは、郵便番号が5桁であった時代でも、同じ郵便番号をもつ地名どうし5キロ以内の距離にあったことからわかる。サーバは会員のコンピュータとインターネットを通じて接続される。

#### 【0010】

10

入会時に、各会員から郵便番号を入力してもらう(この郵便番号から経度緯度テーブル経度・緯度を抽出する)。そして、システムによって各会員のレコードに緯度および経度の項目の値(小数点以下4桁。小数点を入れても各合計8桁ずつ)が付与される。これらは、緯度経度テーブルを参照して得られる。検索時にこの緯度経度テーブルを参照する方法も考えられるが、このテーブルはサイズが大きいため時間がかかる。入会時に一度だけこの緯度・経度を会員テーブルのその会員のレコードに入れておけば、その後の処理は高速になる。

#### 【0011】

出会い系の情報検索システムでは、どの会員も検索することができる。検索する会員が会員番号を入力するとその会員の経度・緯度が特定される。検索時は、前述した計算式にしたがって、全会員のレコードの中から、検索者によって与えられた距離の範囲の会員のレコードを選択する。検索する会員のレコードにも緯度経度の値が付与されているので、両者から距離が計算される。また、検索結果にも条件を満足する会員との距離が表示される。この過程を図5のフローチャートに示す。未読の他会員の緯度経度データがある限り、距離を検索する会員との計算し、その距離がd以内にあれば、その会員をリストに加える。

20

#### 【0012】

経度緯度を会員テーブルのレコードに追加する処理および距離を算出する処理は具体的には、ホームページの動作をつかさどるスクリプトプログラム(例えば、プログラム言語Perl)によって実現される。また、データベースは、MySQLなどのデータベース問合せ言語(SQL)で定義し、簡単なPerlプログラムの中から、INSERT, SELECTなどのSQLコマンドを必要に応じ実行させる。設計時に緯度経度テーブルおよび会員テーブルはCREATEコマンドで定義しておく必要がある。図3にその記述例を示す。2番目に定義したテーブルKaiinのzip, ido, keido以外の項目に何を用いるかは任意である。

30

#### 【0013】

(地表面)距離を求める計算は、Perlの組み込み関数を用いて容易に計算できる。図4にそのコーディング例を示す。

#### 【0014】

本方式を適用する場合は、まず2会員の距離が小さすぎる場合には、検索結果を「10キロ以下」などとして、両会員に互いの住所が近いことをわからせないようにしてもいい。もし、検索結果が「距離が1キロ」と出力されれば、その検索した会員が家の周辺をくまなく探して、相手の会員の住所を突き止める可能性がある。

40

#### 【0015】

##### 【発明の効果】

以上の説明から、本発明は次のような効果を奏する。

まず、請求項1に記載された情報検索では、他会員との距離情報を必要とする会員にとって、より正確な情報を与えることが可能となった。従来は、他会員の住所をばかして与えるだけであったが、この発明により、他会員の住所を隠蔽しながら、真に有用な検索結果(その会員との距離が近いかな否か)を明確に与えることが可能となった。

50

## 【0016】

請求項2に記載された実現方法は、出会い系サイトなど、適用対象によっては低コストで実現できるため、広く利用されるものと思われる。従来からこのような便利な情報検索は、アイデアとして全くなかったわけではない。しかし、低コストで実現する方法がなかったために、絵に描いた餅の域を出なかった。今回の発明により、その具体的な方法が提示されたことになる。

## 【図面の簡単な説明】

【図1】緯度経度テーブルの各レコードの形式を模式的に示す図である。

【図2】会員テーブルの各レコードの形式を模式的に示す図である。

【図3】SQL言語でテーブル定義する命令の記述例を示す図である。

【図4】地表面距離を計算するPerlプログラムコーディング例を示す図である。

【図5】会員の中から自分との距離がd以内の会員を探してリストに入れる処理を模式的に示す図である。

10

## 【図1】

郵便番号	緯度	経度
(十一、7桁)	(小数点以下4桁、全体8桁)	(小数点以下4桁、全体8桁)

## 【図2】

会員番号	他の会員に属する項目たち	緯度	経度
(十一)		(小数点以下4桁、全体8桁)	(小数点以下4桁、全体8桁)

## 【図3】

```
CREATE TABLE idokeido (
  zip int(7) DEFAULT '0' NOT NULL,
  ido decimal(8,4) DEFAULT '0.0000' NOT NULL,
  keido decimal(8,4) DEFAULT '0.0000' NOT NULL,
  PRIMARY KEY(zip)
);
CREATE TABLE kaiin (
  Kaiinbangou char(12) DEFAULT 'Kaiinbangou' NOT NULL,
  Password char(12) DEFAULT 'Password' NOT NULL,
  name char(30) DEFAULT 'name1' NOT NULL,
  EmailAddress char(50) DEFAULT 'EmailAddress' NOT NULL,
  zip char(7) DEFAULT 'zip' NOT NULL,
  address char(200) DEFAULT 'address3' NOT NULL,
  phone char(20) DEFAULT 'phone' NOT NULL,
  ido decimal(8,4) DEFAULT '0.0000' NOT NULL,
  keido decimal(8,4) DEFAULT '0.0000' NOT NULL,
  PRIMARY KEY(Kaiinbangou)
);
```

## 【図4】

```
sub d{
  local($f1,$f2,$g1,$g2,$f1r,$f2r,$g1r,$g2r,$n1,$n2,$a,$e,$r,$swr,$w);
  $a=6378136.0;
  $e=0.006694470;
  $pi=3.141592653589793;
  $f1r=$_[0]*$pi/180.0;
  $f2r=$_[1]*$pi/180.0;
  $g1r=$_[2]*$pi/180.0;
  $g2r=$_[3]*$pi/180.0;
  $n1=$a/(sqrt(1.0-$e*sin($f1r)*sin($f1r)));
  $x1=$n1*cos($f1r)*cos($g1r);
  $y1=$n1*cos($f1r)*sin($g1r);
  $z1=$n1*(1.0-$e)*sin($f1r);
  $n2=$a/(sqrt(1.0-$e*sin($f2r)*sin($f2r)));
  $x2=$n2*cos($f2r)*cos($g2r);
  $y2=$n2*cos($f2r)*sin($g2r);
  $z2=$n2*(1.0-$e)*sin($f2r);
  $r=sqrt(($x1-$x2)*($x1-$x2)+($y1-$y2)*($y1-$y2)+($z1-$z2)*($z1-$z2));
  $wr=atan2($r/2.0/$a,sqrt(1-($r/2.0/$a)*($r/2.0/$a)));
  return(int($a*2.0*$wr/1000));
}
```

【図 5】

